

SUNGARD HIGHER EDUCATION

BANNER GENERAL COMMON MATCHING HANDBOOK

Release 8.4
December 2010

Trademark, Publishing Statement and Copyright Notice

SunGard or its subsidiaries in the U.S. and other countries is the owner of numerous marks, including "SunGard," the SunGard logo, "Banner," "PowerCAMPUS," "Advance," "Luminis," "DegreeWorks," "fsaATLAS," "Course Signals," and "Open Digital Campus." Other names and marks used in this material are owned by third parties.

© 2006-2010 SunGard. All rights reserved.

Contains confidential and proprietary information of SunGard and its subsidiaries. Use of these materials is limited to SunGard Higher Education licensees, and is subject to the terms and conditions of one or more written license agreements between SunGard Higher Education and the licensee in question. This PDF is certified for use with Adobe Reader, version 6.x and higher. Some elements of this PDF may not render properly when viewed using earlier versions of the Acrobat Reader or with other PDF viewing applications.

In preparing and providing this publication, SunGard Higher Education is not rendering legal, accounting, or other similar professional services. SunGard Higher Education makes no claims that an institution's use of this publication or the software for which it is provided will insure compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization's own choosing.

Prepared by: SunGard Higher Education

4 Country View Road
Malvern, Pennsylvania 19355
United States of America

Customer Support Center Website

<http://connect.sungardhe.com>

Documentation Feedback

<http://education.sungardhe.com/survey/documentation.html>

Distribution Services E-mail Address

distserv@sungardhe.com

Revision History Log

Publication Date	Summary
------------------	---------

December 2010	New version that supports Banner General 8.4 software.
---------------	--

Banner General 8.4 Common Matching Handbook

Contents

Chapter 1	Overview of Common Matching	1-1
	Preventing Duplicate Records	1-1
	Common Matching in Banner Forms	1-3
	Data Entry Forms that Use Common Matching	1-3
	Special Processing for the One-Time Payment Form (PEA1PAY)	1-3
	The Online Matching Process	1-3
	Common Matching and Batch Processes	1-6
	Batch Processes that Use Common Matching	1-7
	Common Matching APIs	1-8
Chapter 2	Setting up Common Matching	2-1
	Step 1 Create Matching Source Codes on GTVCMSC.	2-1
	Step 2 Define Source Rules on GORCMSC.	2-2
	Step 3 Define Data Elements on GORCMDD	2-3
	Step 4 Define Name Translations.	2-11
	Step 5 Create Matching Rules on GORCMRL	2-12
	Step 6 Set up User Defaults on GORCMUS.	2-15
	Step 7 Turn on Common Matching	2-16
	Common Matching Forms	2-16
Chapter 3	Using Common Matching	3-1
	Components of Common Matching	3-1
	The Algorithm	3-1
	Matching Source	3-1
	Matching Rules	3-2

Two-Step Matching Process	3-2
Primary Match Processing	3-3
Secondary Match Processing	3-5
Matching Using Multiple Rule Priorities	3-6
Multiple Rule Priorities in Online Processing	3-7
 How Common Matching Creates and Updates Records	 3-8
Address Record	3-9
Telephone Record	3-9
E-mail Record	3-10
Biographical Record	3-10
The Status Message	3-11

Chapter 4 Implementation Examples **4-1**

Sample Common Matching Scenarios	4-1
Person Search That Results in Many Potential Matches	4-1
Person Search That Results in One Match and Potential Matches	4-2
Person Search That Results in an Exact Match	4-4
Status is New but Potential Matches Exist from a Prior Rule	4-5
Match Results That Show a Value Missing	4-6
Data Elements Required but Not Provided	4-7
Data Elements Set as Exists Are Provided and Do Not Match	4-8
Only Exception to the Basic Matching Algorithm	4-9
 Sample Rules	 4-10
Banner Finance Rule for Vendors (Persons)	4-10
Banner Advancement Rule for Persons	4-11
Banner Student Rule	4-13
Banner Financial Aid Rule for Persons	4-14
Sample Banner Advancement Scenarios	4-15
Advancement Rule For Person Matching	4-15
Advancement Person Examples	4-17
Advancement Rule For Non-Person Matching	4-19
Advancement Non-Person Examples	4-20
 Converting Legacy Data - FGAC and Common Matching	 4-22



Integrating Common Matching into Site-Specific Processes 4-22
Step 1 Insert Data into GOTCMME 4-22
Step 2 Execute the Common Matching Procedure 4-23
Step 3 Process the Results of the Common Matching Procedure 4-24

Invoking Common Matching from Additional Forms 4-24
Temporary Access to Restricted Records 4-25

Index I-1



1 Overview of Common Matching

Common Matching is a process that helps your institution create and manage *basic person* records. *Basic person* is a generic term that refers to people (persons) and companies (non-persons). Basic person information includes:

- Name/ID
- Address
- Telephone
- E-mail
- Biographical information

The Common Matching process checks for existing identification records *before* a new one is added to the database. Common Matching also provides a mechanism to add new records to the Banner database and update existing ones. The Common Matching process is used for batch processing and may be used for online matching.

Common Matching replaces the existing matching processes in Financial Aid and Student. For more information on this change, please refer to the Banner Financial Aid 7.0 Release Guide and the Banner Student 7.0 Release Guide.

Only users who have permission to create new identification records in Banner (users who have insert access into the SPRIDEN table) can use Common Matching.

The Common Matching feature is available in forms that create identification forms, but it does not change the existing search features in these forms. See [“Data Entry Forms that Use Common Matching” on page 1-3](#) for a list of forms.

Preventing Duplicate Records

Common Matching uses an algorithm and rules you set up to determine which Banner records might be a duplicate of the one being entered. (For details, see [“Components of Common Matching” on page 3-1](#).)

There are three possible results:

1. The record is *new*. No match has been found on the database. The record can be created without any additional processing, and Banner will assign a new PIDM to it.

Banner uses the new Basic Person APIs to add the information to the database. Users do not need to enter the information again.

2. A *match* is found for the record. Common Matching has found one, and only one, Banner record that matches the record based on the rules. The record appears on the Match tab of GOAMTCH. The user must review the displayed data to see if the matched Banner record is the same as the one they are trying to enter.
 - If the user determines that the record found in the database as a match is the same as the one being entered, the user can select the person or non-person as a Match, or update the record with additional information. In this case, Banner will not assign a new PIDM.

 **Note**

You can only update fields on an existing Banner record if the fields are null in the Banner database. If data already exists for those fields, the existing data will not be overwritten. ■

- If the user determines that the record found as a match by the Matching Process is actually *not* a match, the user can choose to create a new record. Banner will assign a new PIDM to the record when it is saved. Banner uses the new Basic Person APIs to add the information to the database. Users do not need to enter the information again.
3. The new record gets a status of *suspense* because a *potential match* is found. Common Matching has found at least one record where some of the fields identified in the rule match the record being entered, but not all. Potential Matches appear on the Potential Matches tab of GOAMTCH. The number of potential matches found appears on the tab. The potential matches are listed in order by rule set, but can be sorted using the arrows above the ID or name columns.

The user can review each potential match to determine if one is, in fact, a match.

 **Note**

If a record matches based on multiple rules, the record will be displayed multiple times on the Potential Matches tab. The Rule Set Number on which it matched and details of why it matched display when each record is highlighted. ■

- If the user determines that one of the potential matches is the same as the one being entered, the user can select the record as a match or update it with additional information. Banner will not assign a new PIDM. If the record is updated, existing data will not be overwritten.
- If the user determines that none of the potential matches is the same as the one being entered, the user can create a new record. Banner will assign a new PIDM to the record. Banner uses the new Basic Person APIs to add the information to the database. Users do not need to enter the information again.

When new records are created or changes have been made to existing records in the database, a detailed status message is displayed in a pop-up window.

Common Matching in Banner Forms

Data Entry Forms that Use Common Matching

The following data entry forms use Common Matching:

- General Person Identification Form (SPAIDEN)
- Advancement Identification Form (APAIDEN)
- Identification Form (PPAIDEN)
- Personal Identification - Finance Form (FOAIDEN)
- Financial Manager Maintenance Form (FTMFMGR)
- Vendor Maintenance Form (FTMVEND)
- Agency Code Maintenance Form (FTMAGCY)
- Bank Code Rules Form (GXR BANK)
- One-Time Payment Form (PEA1PAY)
- Quick Entry Form (SAAQUIK)
- Quick Recruit Form (SRAQUIK)

Special Processing for the One-Time Payment Form (PEA1PAY)

As part of this form's processing, it enters an institution-wide default value for address type.

If Common Matching is enabled for your institution and the user entering data on this form is not exempt, a person entering a new identification record on this form will be taken to GOAMTCH, as usual. If the user enters a different address type from the default and creates the new record, *the address type brought back into PEA1PAY is the institution default, not the newly-entered one*. The address type the user entered on the GOAMTCH form is not saved in the database.

The Online Matching Process

Note

The following steps assume you have already set up the forms that support the Common Matching process and have selected the **Online Matching Process Enabled** checkbox on GUAINST. ■

GOAMTCH can be accessed directly or is automatically called when:

- A user enters an ID that does not already exist
- A user selects the **Generated** icon to create the next one-up ID
- A user enters the word *GENERATED* in the **ID** field
- A user selects Common Matching (GOAMTCH) from the Options menu on the forms where IDs can be created

To enter a new ID using Common Matching:

1. Access the form on which you want to enter the person or non-person data. Usually this will be one of the %IDEN forms, but you can also enter records on GOAMTCH itself.
2. Enter the ID to be created, or select the **Generated** icon (or enter the word *GENERATED* in the **ID** field) so Banner will use the next available ID.
3. Perform a Next Item or Next Block function. GOAMTCH appears automatically, unless you have been made exempt from the Common Matching process. If you entered an ID to be created, it will automatically appear in the **ID** field.
4. If a default Matching Source code was assigned to your user ID on the Common Matching User Setup Form (GORCMUS), it will automatically appear in the **Matching Source** field on GOAMTCH. If the **Allow Other Matching Sources** checkbox is selected for your user ID on GORCMUS, you can change the Matching Source code on GOAMTCH in order to use a different set of Common Matching rules.

If a default Matching Source code was not assigned to your user ID, enter a Matching Source code in the **Matching Source** field, or select one from the list of values.

5. Perform a Next Block function.
6. Enter information about the person or non-person in the Data Entry window. This information will be used to check for a match. You must enter the **Last Name** for a person, and the **Non-Person** name for a non-person.
7. Select the **Duplicate Check** icon or perform a Next Block function to initiate the Common Matching process. Banner will run the Matching algorithm and process the rules associated with the chosen Matching Source. According to the algorithm and the rules, the system will display any matches or potential matches, or indicate that the record is considered *new*. There are three possible outcomes:
 - 7.1. *New* - The record does not exist on the database, based on the rules associated with that Matching Source. A pop-up window will appear, asking you if you

want to create the record. You can select *Yes* to create the new record, or you can select *No* and then click on the **Create New** icon.

- 7.2.** *Match* - One record matches the data you entered exactly. The record does exist in the database, based on the rules associated with that Matching Source. The information on the matched record appears on the Match tab on GOAMTCH. You can:

Select the **Select ID** icon to bring the information on the matched record back to the form where you started (usually an %IDEN form).

OR

Select the **Update ID** icon if you entered information in GOAMTCH that should be added to the records in the databases. GOAMTCH will attempt to insert or update records in the SPRADDR (address), SPRTELE (telephone), SPBPERS (biographical), and GOREMAL (e-mail) tables. A status message will appear, indicating what data has been created or updated. You will then be returned to the form where you started.



Note

You can only update fields on an existing Banner record if they are null in the Banner database. If data already exists for those fields, it will not be overwritten. ■

- 7.3.** *Potential Matches* - Based on the rules for that Matching Source, more than one record matches the data you entered, or there are multiple records where some of the information is matched, but not all of it. For example, it could be that the first name, last name, and address you entered exist in the database but the date of birth is different, or that two records with the same first name, last name, address and date of birth are found. The choices are listed on the Potential Matches tab on GOAMTCH.



Note

The potential matches can be sorted dynamically by ID or name in ascending or descending order using the Sort up and down arrow buttons for the ID and Name fields. The default sort order is by ID and priority (in descending order). ■

When you click on the Sort up arrow, the sort organizes the results in alpha order (A - Z). When you click on the Sort down arrow, the sort organizes the results in reverse alpha order (Z - A).

Once you have clicked on an up or down arrow, you can mouse over the arrow to see the tool switch hint message, click on the arrow again, reverse its direction, and perform a new sort. For example, after you have sorted on ID or

name in A - Z order (using the Sort up arrow), you can then click on the arrow again to change it to a Sort down arrow, and resort the data in Z - A order.

Select the **Details** icon to see a list of forms that may be helpful in determining if the highlighted record is a match for the one you're entering. Select the form name from the list to access the form.

If the record you're adding is not listed (none of the potential matches are the same as the record you are creating), select the **Create New** icon to create a new record in Banner. If one of the potential matches is the same as the record you're adding, select **Select ID** or **Update ID** as described above for the highlighted record on the Potential Matches tab.

8. Continue entering data on the original form.

 **Note**

Unless you are exempt or your institution has elected not to use Common Matching, you cannot bypass the Common Matching process. If you do not select one of the options after initiating the search for matches, you cannot perform a Next Block function to enter any data after you return to the original form. ■

If the record already exists, it can be updated with information from the new record in the following circumstances, but a new PIDM will not be created:

- SSN/SIN/TIN if it is null in the database
- Birth date if it is null in the database
- Gender if it is unknown in the database
- Address, telephone, and e-mail if the type you entered on GOAMTCH does not already exist for the record
- A new sequence number will be created for the address if it already exists in the database with the same type but the address information is different

 **Note**

If an address record is created and an active address already exists for the same Address Type, the original address will be made inactive. ■

Common Matching and Batch Processes

You must set up Common Matching to be used for records added via batch processes as well as records entered via online processing from the data load view forms.

The batch processes have been changed to use the new Common Matching algorithm and its associated source codes. Each batch process will continue to function as it did before

this release, but users will now use GOAMTCH to resolve the potential matches that are found.

Upon completion, each process will have done the following based on the rules you set up for each data source:

1. Created or identified new records as necessary, depending on your process
2. Matched incoming records to existing Banner records
3. Marked records as SUSPENDED (when there are potential matches)

Use the existing Banner forms to match records or to resolve suspended records (e.g., use SAAEAPS to resolve Web Admissions applications). These forms now call GOAMTCH when the user selects the Match or Associate ID option.

Incoming records are stored in temporary tables. When users use a form to match data loaded by a batch process, GOAMTCH is called. The Data Entry block of GOAMTCH will be populated with the information from the temporary tables. Users can perform a Next Block to run the Common Matching algorithm, and, based on the results, they can determine if the record is new or if it is the match of an existing Banner record. They will then be returned to the calling form.

For more information about using Common Matching with specific batch processes and the related data load review forms, please refer to the product-specific release guides.

Batch Processes that Use Common Matching

The following batch processes use Common Matching:

- Electronic Application Verify/Load Process (SARETMT)
- Electronic Prospect Match Process (SRRSRIN)
- Financial Aid Data Load Part 2 Process (RCPMTCH)

The following batch process review forms use Common Matching:

- Electronic Application Process Form (SAAEAPS)
- Electronic Prospect Inquiry Form (SRIPREL)
- Online Transcripts Activity List Form (SHAEDIS)
- Financial Aid Suspended Record Maintenance Form (RCRSUSP)

Common Matching APIs

Several Banner APIs support the common matching process. The form listed next to the table in this chart is the representative source used to build the API validation and business rules.

Table	Form	API Object Name	API Entity Name	Task Performed
GORCMDD	GORCMDD	gb_cm_data_dictionary	COMMON MATCHING DATA DICTIONARY	Maintains data dictionary entries for common matching
GORCMDO	GORCMRL	gb_cm_display_options	COMMON MATCHING DISPLAY OPTIONS	Maintains options for how common matching search results will be displayed
GORCMSR	GORCMRL	gb_cm_rules	COMMON MATCHING RULES	Maintains common matching rules
GORCMSP	GORCMRL	gb_cm_source_priority	COMMON MATCHING SOURCE PRIORITY	Maintains priority numbers for common matching source codes
GORCMSC	GORCMSC	gb_cm_source_rules	COMMON MATCHING SOURCE RULES	Maintains source codes for common matching
GORCMUP	GORCMRL	gb_cm_user_procedure	COMMON MATCHING USER PROCEDURE	Associates packaged procedures to be used by the common matching procedure
GOBCMUS	GORCMUS	gb_cm_user_setup	COMMON MATCHING USER SETUP	Maintains users for common matching

2 Setting up Common Matching

To set up Common Matching, perform the following steps.

1. Create Matching Source codes on the Common Matching Source Code Validation Form (GTVCMSC) on page [2-1](#).
2. Define Source Rules on the Common Matching Source Code Rules Form (GORCMSC) on page [2-2](#).
3. Define Data Elements for matching on the Common Matching Data Dictionary Form (GORCMDD) on page [2-3](#).
4. Define Name Translations on the Name Translation Rules Form (GORNAME) and/or Non-Person Name Translation Rules Form (GORNPNM) on page [2-11](#).
5. Create Matching Rules on the Common Matching Rules Form (GORCMRL) on page [2-12](#).
6. Set up User Defaults on the Common Matching User Setup Form (GORCMUS) on page [2-15](#).
7. Turn on Common Matching on the Installation Control Form (GUAINST) on page [2-16](#).

 **Note**

The GUAINST setting affects only online Common Matching. Common Matching is always used during batch processes, regardless of your choices on GUAINST. ■

Step 1 Create Matching Source Codes on GTVCMSC

Common Matching uses rules based on the source of the data (for example, ACT test scores for Student, EDE data load for Financial Aid, and online data entry into APAIDEN for Advancement). Each Matching Source Code can have multiple rules that will be used by the Matching Algorithm to determine if the record being entered already exists in the

database. Different organizations or departments within your institution can set up different Matching Source Codes and associated rules based on their specific needs.

1. Go to the Common Matching Source Code Validation Form (GTVCMSC).
2. Define the Common Matching source codes.



Tip

Use naming conventions and detailed descriptions. ■

3. (Optional) Create a comment in the **Comment** field.
4. Save.

Step 2 Define Source Rules on GORCMSC

After setting up Matching Source codes, you must define the rules that will be used with each source code.

1. Go to the Common Matching Source Code Rules Form (GORCMSC).
2. Choose a **Matching Source** code.
3. (Optional) Check the **Transpose Date Month / Day** check box if you want Common Matching to look for a match on the full birth date. If one is not found, it will then look at just the month and the day, and then transpose the month and day. For example, if you enter *01/05* on GOAMTCH, you will receive matches on January 5th and May 1st.



Note

You must include the **Birth Date** element in your rule in order to use this option. ■

4. (Optional) Check the Transpose **First Name / Last Name** check box. This option allows for the first and last names to be transposed. For example, Thomas Lee = Lee Thomas.
5. (Optional) Check the **Allow Alias Wildcard Use** check box. This option appends a % (percent wildcard character) to incoming data in order to search for similar rows in alias tables. For example, if you enter *Bet* on GOAMTCH, Common Matching will find Beth = Elizabeth in the alias table and return potential matches with a first or middle name of Elizabeth.
6. (Optional) Check the **Allow Length Override** check box. This option uses the length of incoming data to search for matches on name fields, if the length of data on the rule is greater than the data input. For example, assume that you have a rule that specifies a length of 5 for First Name. With the **Allow Length Override** option, if

you enter Sam in First Name field on GOAMTCH, then potential matches that begin with Sam, such as Samuel, Samantha, and Samson will be found as potential matches.

7. (Optional) Check the **Prevent ID Creation on API Failure** check box. If you want to use this option, you must provide complete information for Address, Telephone and/or E-mail in order to create the new ID. You can proceed with ID creation by removing partial information.

 **Note**

This option does not affect batch loads. ■

8. (Optional) Check the **Create Hierarchy of Display Using Defaults** button to populate the **Address**, **Telephone**, and **E-mail** fields in the Hierarchy of Display blocks (GORCMDH table) from the Data Entry / Update Defaults values (GORCMSC table).
9. (Optional) Specify default address, telephone, or e-mail types.

 **Note**

These default values will appear on the Common Matching Entry Form (GOAMTCH) when a user is entering data to match. When the user chooses to create a new identification record or to update an existing one, the default values are used in the new or updated record. The default values are not used by Common Matching to determine a match. ■

10. If you want the Matching Source code to be able to use Common Matching batch elements (those that do not have the **Online** indicator checked), then leave the **Use for Online Common Matching** check box unchecked. This allows batch elements of the Matching Source code to be used during batch processing only. If you do check the **Use for Online Common Matching** check box, then when you are building your rules in the next step, you will be limited to only using elements in your matching rule that the user can input on GOAMTCH. These are the elements that have the **Online** check box checked on GORCMDD.
11. Specify whether the source code will be used to match against person and/or non-person data by choosing a value for the **Match Type** field.
12. (Optional) Specify any forms that the user can access from GOAMTCH to see the details for potential matches. You may enter up to 6 forms and assign a sequence number for each, which determines the order in which the forms will be displayed when the user selects the **Details** button on GOAMTCH.
13. Save.

Step 3 Define Data Elements on GORCMDD

The Common Matching Data Dictionary Form (GORCMDD) displays the data elements you can use in the Common Matching process. These data elements are the building blocks for your rules.

This form is delivered with available data elements that your institution can use, and you can also add data elements to use with package procedures developed by your institution.

Using Site-Specific Elements in the Data Dictionary

You can add your own site-specific data elements to GORCMDD by using the following steps.

1. Define the data elements.

Define the data elements you want to include in the Common Matching procedure. Are they all on the same table? Do they represent the same entity?

For example, if you're adding high school information you could have:

```
SORHSCH_SBGI_CODE  
STVSBGI_DESC  
SOBSBGI_CITY  
SOBSBGI_STAT_CODE  
SOBSBGI_ZIP
```

All these columns pertain to the same entity. You might want to create a view to use with Common Matching, to make it easier to update the data dictionary (GORCMDD), the rules, and subsequent cursor.

2. Create or change a temporary table.

Data that will be added to Banner tables after being checked for existing records is stored in a global temporary table during the Common Matching process.

- If a table already exists for your product, add your new elements to that table. Banner Student is delivering the SOTCMME table for this release:

SOTCMME_SBGI_CODE	VARCHAR2(6)
SOTCMME_SBGI_NAME	VARCHAR2(30)
SOTCMME_SBGI_CITY	VARCHAR2(20)
SOTCMME_SBGI_STAT_CODE	VARCHAR2(3)
SOTCMME_SBGI_ZIP	VARCHAR2(10)

- If one does not exist, use the Common Matching Entry Table (GOTCMME) as a template to create one.

GOTCMME is an Oracle global temporary table. The data it contains is specific to a particular session. When the session is finished, the rows it contains are deleted. More than one session can use GOTCMME, and each session can only see the rows it created.

3. Add the elements to the Data Dictionary.

Add the new elements to the data dictionary on the Common Matching Data Dictionary Form (GORCMDD).

For high school information, the elements could look like this:

Base Table	Column Name	Element	Maximum Length	Length Updateable	Allow Negative Length
SORHSCH	SORHSCH_SBG_CODE	High School Code	6	N	N
SORHSCH	STVSBGI_DESC	High School Name	30	Y	N
SORHSCH	SOBSBGI_CITY	High School City	20	Y	N
SORHSCH	SOBSBGI_STAT_CODE	High School State	3	N	N
SORHSCH	SOBSBGI_ZIP	High School ZIP Code	10	Y	N

The elements that represent one entity should have the same table name. The algorithm uses the table name to see if the entity has any rules.

Do not check the **Online** indicator (GORCMDD_ONLINE_MATCH_IND = N) for your site-specific procedures. The columns you enter will not have corresponding fields on GOAMTCH. Users cannot enter a person's name and high school on GOAMTCH to see if a record already exists for that person.

Do not check the **Core Matching Element** indicator (GORCMDD_ELEMENT_REQ_IND = N) because the elements are not required for a Matching Source.

4. Create the new procedure.

Create the new procedure using the following as a template, then:

- 4.1. Change the procedure to select the global temporary table you created in Step 2 instead of the Common Matching Entry Table (GOTCMME). Banner Student delivers sakmtch.p_match_sorhsch to match against high school data fields.
- 4.2. Change the cursor to select the new data elements you defined in Step 1.

```

SET SCAN OFF
--
-- AUDIT TRAIL: 7.0
-- 1. DA 02/09/2004
--   Created package to support the new Common Matching functionality.
--
-- AUDIT TRAIL END
--
CREATE OR REPLACE PACKAGE gokdebt AS
--
PROCEDURE p_match_sorhsch ( p_cmsc_code      IN OUT  gotcmrt.gotcmrt_cmsc_code%TYPE,
                           p_cmsr_priority_no IN OUT  gotcmrt.gotcmrt_cmsr_priority_no%TYPE,
                           p_pidm           IN OUT  gotcmrt.gotcmrt_pidm%TYPE,
                           p_result_type    IN OUT  gotcmrt.gotcmrt_result_type%TYPE,
                           p_match_count    IN OUT  gotcmrt.gotcmrt_match_count%TYPE,
                           p_missing_count  IN OUT  gotcmrt.gotcmrt_missing_count%TYPE,
                           p_unmatch_count  IN OUT  gotcmrt.gotcmrt_unmatch_count%TYPE,
                           p_message        IN OUT  gotcmrt.gotcmrt_message%TYPE );
--
END gokdebt;
/
SHOW ERRORS;
--
SET SCAN ON
WHENEVER SQLERROR CONTINUE;
DROP PUBLIC SYNONYM gokdebt;
WHENEVER SQLERROR EXIT ROLLBACK;
CREATE PUBLIC SYNONYM gokdebt FOR gokdebt;
WHENEVER SQLERROR CONTINUE;
--START gurgrtb gokdebt
--START gurgrti gokdebt
WHENEVER SQLERROR EXIT ROLLBACK;

```

```

SET SCAN OFF
--
-- AUDIT TRAIL: 7.0
-- 1. DA 02/09/2004
--   Created package to support the new Common Matching functionality.
--
-- AUDIT TRAIL END
--
CREATE OR REPLACE PACKAGE BODY gokdebt AS
--
TYPE GenericCurTyp IS REF CURSOR;
--
-- Constants
--
M_SORHSCH          CONSTANT VARCHAR2(32) := 'SORHSCH';
--
M_SORHSCH_SBG_CODE CONSTANT VARCHAR2(32) := 'SORHSCH_SBG_CODE';
--
HSCH_MATCH_MSG     CONSTANT VARCHAR2(32) := g$_nls.get('x','SQL','High School
Match');
HSCH_MISSING_MSG   CONSTANT VARCHAR2(32) := g$_nls.get('x','SQL','High School
Missing');
HSCH_NO_MATCH_MSG  CONSTANT VARCHAR2(32) := g$_nls.get('x','SQL','High School No
Match');
--
FUNCTION f_sorhsch_exists ( p_pidm IN gotcmrt.gotcmrt_pidm%TYPE )
RETURN BOOLEAN ;

```

```

--
PROCEDURE p_open_sorhsch_cv ( p_sorhsch_cv_inout   IN OUT  GenericCurTyp,
                             p_gotcmme_row        IN      gotcmme%ROWTYPE,
                             p_pidm               IN      gotcmrt.gotcmrt_pidm%TYPE,
                             p_cmssc_code         IN      gotcmrt.gotcmrt_cmssc_code%TYPE,
                             p_cmssc_priority_no  IN      gotcm-
rt.gotcmrt_cmssc_priority_no%TYPE );
-----

FUNCTION f_sorhsch_exists ( p_pidm IN  gotcmrt.gotcmrt_pidm%TYPE )
RETURN BOOLEAN IS
--
exists_ind  VARCHAR2(1);
--
CURSOR sorhsch_exists_c IS
SELECT 'Y'
FROM sorhsch
WHERE sorhsch_pidm = p_pidm ;
--
BEGIN
--
OPEN sorhsch_exists_c;
FETCH sorhsch_exists_c INTO exists_ind;
IF sorhsch_exists_c%NOTFOUND THEN
CLOSE sorhsch_exists_c;
RETURN FALSE;
END IF;
--
CLOSE sorhsch_exists_c;
RETURN TRUE;
--
END f_sorhsch_exists;
-----

PROCEDURE p_match_sorhsch ( p_cmssc_code         IN OUT  gotcmrt.gotcmrt_cmssc_code%TYPE,
                             p_cmssc_priority_no  IN OUT  gotcmrt.gotcmrt_cmssc_priority_no%TYPE,
                             p_pidm               IN OUT  gotcmrt.gotcmrt_pidm%TYPE,
                             p_result_type        IN OUT  gotcmrt.gotcmrt_result_type%TYPE,
                             p_match_count        IN OUT  gotcmrt.gotcmrt_match_count%TYPE,
                             p_missing_count      IN OUT  gotcmrt.gotcmrt_missing_count%TYPE,
                             p_unmatch_count      IN OUT  gotcmrt.gotcmrt_unmatch_count%TYPE,
                             p_message           IN OUT  gotcmrt.gotcmrt_message%TYPE ) IS
--
DataRec      gokcmpk.DataRecordTyp;
ResultRec    gokcmpk.ResultRecordTyp;
ResultTab    gokcmpk.ResultTabTyp;
--
result_index BINARY_INTEGER;
--
data_cv      GenericCurTyp;
--
TYPE refcursor IS REF CURSOR;
--
hschdtlcur   refcursor;

```

```

--
result_ind      gotcmrt.gotcmrt_result_type%TYPE;
row_id          VARCHAR2(18);
match_cnt      INTEGER;
missing_cnt    INTEGER;
no_match_cnt   INTEGER;
--
gotcmme_row     gotcmme%ROWTYPE;
--
CURSOR get_gotcmme_c IS
  SELECT *
  FROM gotcmme ;
--
BEGIN
--
  IF NOT gokcmpk.f_table_rule_exists ( p_cmssc_code,
                                      p_cmssr_priority_no,
                                      M_SORHSCH) THEN
    RETURN;
  END IF;
--
  OPEN get_gotcmme_c;
  FETCH get_gotcmme_c INTO gotcmme_row;
  IF get_gotcmme_c%NOTFOUND THEN
    CLOSE get_gotcmme_c;
    RETURN;
  END IF;
--
  CLOSE get_gotcmme_c;
--
  ResultTab.DELETE;
  result_index := 0;
--
  p_open_sorhsch_cv ( data_cv,
                     gotcmme_row,
                     p_pidm,
                     p_cmssc_code,
                     p_cmssr_priority_no );
--
  LOOP
    FETCH data_cv INTO row_id,
              hschdtlcur;
    EXIT WHEN data_cv%NOTFOUND ;
--
    result_index := result_index + 1;
    ResultTab(result_index).result_row_id := row_id;
    ResultTab(result_index).result_row_match_cnt := 0;
    ResultTab(result_index).result_row_missing_cnt := 0;
    ResultTab(result_index).result_row_no_match_cnt := 0;
--
    LOOP
--
      FETCH hschdtlcur INTO DataRec;
      EXIT WHEN hschdtlcur%NOTFOUND ;
--
      gokcmpk.p_match_data ( DataRec,
                           match_cnt,
                           missing_cnt,
                           no_match_cnt );
--
      ResultTab(result_index).result_row_match_cnt := Result-
Tab(result_index).result_row_match_cnt + match_cnt;
      ResultTab(result_index).result_row_missing_cnt := Result-
Tab(result_index).result_row_missing_cnt + missing_cnt;
      ResultTab(result_index).result_row_no_match_cnt := Result-
Tab(result_index).result_row_no_match_cnt + no_match_cnt;
--
    END LOOP;
--
  END LOOP;

```

```

--
-- move the results for the address to the table.
--
IF      ResultTab(result_index).result_row_no_match_cnt = 0 THEN
    ResultTab(result_index).result_row_result_ind := gokcmpk.M_MATCH_IND;
ELSIF ResultTab(result_index).result_row_no_match_cnt = Result-
Tab(result_index).result_row_missing_cnt THEN
    ResultTab(result_index).result_row_result_ind := gokcmpk.M_MISSING_IND;
ELSE
    ResultTab(result_index).result_row_result_ind := gokcmpk.M_NO_MATCH_IND;
END IF;
--
END LOOP;
CLOSE data_cv;
--
-- process the results table and output the final results
--
gokcmpk.p_process_result_table ( ResultTab,
                                result_ind,
                                row_id,
                                match_cnt,
                                missing_cnt,
                                no_match_cnt );
--
IF      result_ind = gokcmpk.M_MATCH_IND THEN
    p_message := p_message || ', ' || HSCH_MATCH_MSG;
ELSIF result_ind = gokcmpk.M_MISSING_IND THEN
    p_message := p_message || ', ' || HSCH_MISSING_MSG;
ELSE
    p_message := p_message || ', ' || HSCH_NO_MATCH_MSG;
END IF;
--
p_match_count := p_match_count + match_cnt;
p_missing_count := p_missing_count + missing_cnt;
p_unmatch_count := p_unmatch_count + no_match_cnt;
--
END p_match_sorhsch ;

-----

PROCEDURE p_open_sorhsch_cv ( p_sorhsch_cv_inout IN OUT GenericCurTyp,
                             p_gotcmme_row      IN      gotcmme%ROWTYPE,
                             p_pidm             IN      gotcmrt.gotcmrt_pidm%TYPE,
                             p_cmssc_code      IN      gotcmrt.gotcmrt_cmssc_code%TYPE,
                             p_cmssc_priority_no IN      gotcm-
rt.gotcmrt_cmssc_priority_no%TYPE ) IS

```

```

--
BEGIN
--
  IF f_sorhsch_exists ( p_pidm ) THEN
    OPEN p_sorhsch_cv_inout FOR
      SELECT ROWIDTOCHAR(y.rowid),
             CURSOR (SELECT DECODE(GORCMSR_COLUMN_NAME,
                                   M_SORHSCH_SBGI_CODE ,
p_gotcmme_row.gotcmme_email_address),
                   DECODE(GORCMSR_COLUMN_NAME,
                           M_SORHSCH_SBGI_CODE , sorhsch_sbgi_code),
                   gorcmsr_column_name,
                   gorcmdd_element,
                   gorcmsr_data_req_ind,
                   decode( sign(gorcmsr_length),1,1,gorcmsr_length ),
                   decode(
sign(gorcmsr_length),1,gorcmsr_length,(gorcmsr_length*-1) )
             FROM   gorcmdd,
                   gorcmsr,
                   sorhsch z
             WHERE  gorcmdd_table      = M_SORHSCH
                   AND gorcmdd_column_name = gorcmsr_column_name
                   AND gorcmsr_cmesc_code = p_cmesc_code
                   AND gorcmsr_priority_no = p_cmesc_priority_no
                   AND y.rowid        = z.rowid )
      FROM sorhsch y
      WHERE y.sorhsch_pidm = p_pidm ;
  ELSE
    OPEN p_sorhsch_cv_inout FOR
      SELECT ' ',
             CURSOR (SELECT DECODE(GORCMSR_COLUMN_NAME,
                                   M_SORHSCH_SBGI_CODE ,
p_gotcmme_row.gotcmme_email_address),
                   DECODE(GORCMSR_COLUMN_NAME,
                           M_SORHSCH_SBGI_CODE , '' ),
                   gorcmsr_column_name,
                   gorcmdd_element,
                   gorcmsr_data_req_ind,
                   decode( sign(gorcmsr_length),1,1,gorcmsr_length ),
                   decode(
sign(gorcmsr_length),1,gorcmsr_length,(gorcmsr_length*-1) )
             FROM   gorcmdd,
                   gorcmsr
             WHERE  gorcmdd_table      = M_SORHSCH
                   AND gorcmdd_column_name = gorcmsr_column_name
                   AND gorcmsr_cmesc_code = p_cmesc_code
                   AND gorcmsr_priority_no = p_cmesc_priority_no )
      FROM dual ;
  END IF;
--
END p_open_sorhsch_cv ;

-----
--
--
END gokdebt;
/
SHOW ERRORS;
--
SET SCAN ON

```

5. Add the new elements to the appropriate rules.

Add the elements to the Common Matching rules on the Matching Rules tab of the Common Matching Rules Form (GORCMRL).

6. Create the new procedure.

Add the procedure to the Common Matching rules on the Matching Procedures tab of the Common Matching Rules Form (GORCMRL).

 **Note**

If the procedure does not have a synonym, be sure to include the owner, e.g., `baninst1.gokdebt.p_match_sorhsch`. ■

7. Change existing processes to insert data into the new temporary table.

Change existing processes as necessary to insert values into the temporary table you created or changed in Step 2. Any process that includes Common Matching and could use the new rules must be changed to perform the inserts *before* calling Common Matching.

Banner Student programs SARETMT and SRRSRIN and the package SAKQADM have been changed to insert into the new SOTCMME table so you can match on high school fields that you could match on in 6.x. The forms SAAEAPS and SRIRPREL will also allow you to use SOTCMME data fields for matching.

8. Test.

Test the process and rules using the matching procedure you created on GORCMRL. Make sure the correct messages appear for the new data elements.

Step 4 Define Name Translations

You must set up “aliases” (or name translations) for the names of people and non-persons (such as companies). When you set up aliases, the Common Matching process can determine that records with either a matching name (such as Jonathan) or alias (such as Jon) belong to the same entity.

The translations work both ways. You do not need to create one record associating, for example, *William* with *Bill* and a second record associating *Bill* with *William*. One record with the combination will suffice.

 **Note**

You cannot change the name/alias combinations after you save them on this form. You must delete the record and enter it again. ■

For a Person

GORNAME is used for first and middle names, and is delivered with data for common names. You must add additional data to fit the needs of your site. A script is delivered to migrate data from the SORNAME table in Student to the GORNAME table.

1. Access the Name Translation Rules Form (GORNAME).
2. Enter the name for which you want to create an alias in the **Name** field.
3. Enter the alias in the **Alias** field.
4. Save.

For a Non-person

You must add data for non-persons to fit the needs of your site.

1. Access the Non-Person Name Translation Rules Form (GORNPNM).
2. Enter the name for which you want to create an alias in the **Name** field.
3. Enter the alias in the **Alias** field.
4. Save.

Step 5 Create Matching Rules on GORCMRL

After defining your source rules, you must create matching rules.

1. Access the Common Matching Rules Form (GORCMRL).
2. Choose a **Source** code. Go to the next block.
3. Enter a priority and description for your first matching rule. Save.

The algorithm processes each rule that has been defined for the Matching Source code separately and completely, based on the priority given in the Rule Priorities block.

If the Rule Set 1 determines that the input record is either New or Matched, the overall status that is returned for the record is New or Matched. No additional rules will be processed.

Note

For online processing, the potential matches that result from processing the rules may be viewed online from the Potential Matches tab on the Common Matching Entry Form (GOAMTCH). Potential matches are listed in order by rule set. ■

4. If you are creating a matching rule based on a Banner table, go to the Matching Rules tab and choose the Data Element you want to match on.

For each data element:

- 4.1. Indicate the length of the item to be used for matching (for example, *l* will match the first character) in the **Length** field.

Whenever a length is specified for a Data Element on the Common Matching Rules form (GORCMRL), the comparison will be made using the rule length of the fields.

For example, using the last name, the comparison will be between the Data Element length of the last name on the incoming record to the Data Element length of the last name in Banner. If the rule length is 5, then the first five characters of the incoming record's last name are compared to the first five characters of the Banner last name.

A negative length may be entered for the ID and SSN/SIN/TIN fields to reverse the order to last to first. For example, if -5 is entered for the length of the ID, the last 5 characters of the ID for the incoming record will be compared to the last 5 characters of the ID stored in Banner.

SunGard Higher Education recommends that you do not use a negative length for SSN/SIN/TIN because it could create performance issues.

Example:

Rule Set 1:

Last Name length: 4

First Name length: 3

ID length: -4

Patricia Longnecker, A00027280

The first 4 characters of the last name would be used: LONG

The first 3 characters of the first name would be used: PAT

The last 4 characters of the ID would be used: 7280

 **Note**

SPRIDEN_SEARCH_LAST_NAME always needs to be required. Although it is possible to set up a primary rule on GORCMRL with the **Match on Null--Yes or No** field for SPRIDEN_SEARCH_LAST_NAME set to **Yes**, this is not supported or recommended, because doing so may cause incorrect results on GOAMTCH.

Since primary match logic uses last name/non-person name (SPRIDEN_SEARCH_LAST_NAME), a required data element for Common

Matching, you must set the **Match on Null--Yes or No** field for SPRIDEN_SEARCH_LAST_NAME to *No*. ■

 **Note**

Negative length on GORCMRL--Baseline Common Matching can only handle negative GORCMRL length for two fields, SPBPERS_SSN and SPRIDEN_ID. Negative lengths are not supported for any other baseline Common Matching fields. SunGard Higher Education recommends that you do not use a negative length for SSN/SIN/TIN because it could create performance issues. ■

- 4.2. In the **Match on Null Data – Yes or No** field, choose the **Yes** button to indicate that Banner and/or incoming data can be NULL or identical to be considered a Match. Choose the **No** button to indicate that Banner and incoming data must be NOT NULL and identical to be considered a Match.

 **Note**

If your only rules are built using Name and SSN/SIN/TIN only and the match is not exact, the record will be marked as New. In order to cause the same record to suspend and potentials to be displayed, you need to include at least one more non-null and non-matched "No" data element or one non-matched "Yes" data element in your rule. ■

5. If you are creating a matching rule that is not based within a Banner table, then go to the Matching Procedures tab and add the specific database procedure(s) for matching components, and assign a priority to each one.

 **Note**

On this form, you can also assign site-specific procedures as rules for fields that do not reside in Banner General. For example, Banner Student allows you to use high school information to determine if an identification record being created already exists on the database. A matching procedure, SAKMTCH.P_MATCH_SORHSCH, has been delivered with this release. It allows the source code to use fields associated with high school data in the matching algorithm. ■

6. (Optional) Go to the Copy Rules To tab and copy your rule to another existing Rule Set that does not have any rules yet.
7. Save.

Step 6 Set up User Defaults on GORCMUS

Next, you can control how Common Matching is used by Banner users on the Common Matching User Setup Form (GORCMUS). This form allows you to:

- Specify a default Matching Source for a Banner user
- Allow a user to use other Matching Sources
- Exempt a user from the Common Matching Process

Note

If you do not create a record for a user on GORCMUS, and that user has Common Matching enabled for him/her, then he/she will be able to choose any available Matching Source when on the GOAMTCH form. The Matching Source chosen will be retained for the remainder of the Banner session. ■

1. Access the Common Matching User Setup Form (GORCMUS).
2. Insert a record and enter a valid Banner user (Oracle User ID).
3. Choose a default online Matching Source code for the user for online matching.
4. Check the **Allow Other Matching Sources** check box to indicate that the user is allowed to choose from other Matching Source codes. Leave the check box unchecked to restrict the user to only using the default Matching Source.
5. If you want to exclude the user from Common Matching, check the **Exclude User** check box.

Note

If users are not defined on this form, Common Matching will be enabled for them but they will not have a default source code (unless a system wide default has been established as described in the next section) the first time they enter the GOAMTCH form. However, the source code they choose will be retained for them during their current session. ■

6. Save.

Set up a System Wide Default Matching Source Code

The following instructions describe how to set up a system wide default matching source code rule on the GTVSDAX.form.

1. Enter *CM_SOURCE_CODE* in the **Group** field.
2. Enter *DEFAULT* in the **Code** field.

3. Enter the system wide default Matching Source Code Rule in the **External Code** field.

 **Note**

The maximum length of this rule is limited to 15 characters. If a User Default does not exist and the GTVSDAX entry does exist, then that value will be used as the user default. This enables an institution to provide a default for everyone without creating individual GORCMUS rules for every user. ■

If no system wide default is required, you can do either one of the following:

- Delete the GTVSDAX entry.
- Leave the GTVSDAX **External Code** field blank or *<UPDATE ME>* since those values are treated as non-existing as well.

Step 7 Turn on Common Matching

The final setup step for implementing Online Common Matching is to enable it, so users will automatically be taken to GOAMTCH when creating a new IDs on %IDEN forms, or on any other forms that use Common Matching.

 **Note**

Batch Processes use Common Matching regardless of whether Online Common Matching is enabled on GUAINST. ■

1. Access the Installation Controls Form (GUAINST).
2. Check the **Online Matching Process Enabled** check box.
3. Save.

 **Note**

If you do not enable Common Matching at the site level, individual users can still use it by accessing GOAMTCH directly. ■

Common Matching Forms

Information on Banner forms and fields can be found in Banner Online Help. The help system runs in a web browser, and is available by choosing the **Online Help** button on the toolbar or the Online Help option from the Help pull-down menu. Additionally, you can access online help by selecting the **Help Center** link on the main menu.

Beginning with Release 8.0, Online Help includes a PDF version of form and field reference, which you can print or save to your local computer.

The following Banner forms are used for Common Matching.

Common Matching Source Code Validation Form (GTVCMSC)

This form allows you to set up a code that represents each source of new person and non-person records.

Common Matching Source Rules Form (GORCMSC)

This form allows you to associate a matching source with default information. A corresponding source code must already exist on GTVCMSC.

Common Matching User Setup Form (GORCMUS)

This form allows an administrative user to assign a default Common Matching source code to users for online processing.

Common Matching Data Dictionary Form (GORCMDD)

This form allows you to review the data elements you want to use in the Common Matching process. The data dictionary contains all the available data elements that may be used in Common Matching rules and information about those elements.

Common Matching Rules Form (GORCMRL)

This form allows you to set up the rules that the Common Matching process will use to search the database to see if a person or non-person record already exists when a new record is being added to the database.

Common Matching Entry Form (GOAMTCH)

This form appears automatically when a user tries to add a new identification record to the Banner database, unless the user has been excluded from the Common Matching process, or the Online Matching Processing Enabled checkbox is cleared on the Institution Controls Form (GUAINST). Excluded users can still access GOAMTCH directly, or by selecting it from the Options menu while on an %IDEN form.

Non-Person Name Translation Rules Form (GORPNPM)

This form allows you to set up rules that the Common Matching process will use when searching to see if a record already exists for the specified non-person. This would be helpful if, for example, a company is commonly known by its initials. The Common Matching process would then recognize records with either the name or the initials as belonging to the same entity.

Name Translation Rules Form (GORNAME)

This form allows you to set up rules that the Common Matching process will use when searching to see if a record already exists for the specified person.

Common Matching Source Search Form (GOICMSS)

Use this form to query for all matching sources for a particular element. You can also use it to quickly update the matching rules at your institution.

Data Source Rules Form (RCRDTSR)

Use this form to establish the Common Matching Source Code and RCRT_{xx} parameter set to use when you process records from the Financial Aid Suspended Record Maintenance Form (RCRSUSP).

3 Using Common Matching

This chapter discusses how to use Common Matching.

Components of Common Matching

The Common Matching process is made up of multiple components.

- The Common Matching Algorithm
- The Matching Source
- The Matching Rules associated with the Matching Source

The Algorithm

When the Common Matching process is invoked, the Matching Algorithm performs a Primary and Secondary Match for each rule associated with the Matching Source to determine if an incoming record matches an existing Banner record. See [“Two-Step Matching Process” on page 3-2](#).

Matching Source

Rules are organized by Matching Source. For example, the Matching Source and associated rules used when records are created by the ACT test score data load can be different than the Matching Source and associated rules used when records are created by online Financial Aid processing. Each Matching Source can have multiple rules assigned to it.

Each Matching Source will have a different code. Certain default values can be assigned to the Matching Source to be used when creating new records. See [“Define Source Rules on GORCMSC” on page 2-2](#).

You can specify which rules will be in force for specific users by assigning a Matching Source code to their user IDs. See [“Set up User Defaults on GORCMUS” on page 2-15](#).

Matching Rules

The rules that define how the Common Matching algorithm searches for duplicates are defined on the Common Matching Rules Form (GORCMRL). There are certain restrictions on these rules:

- The algorithm requires that the last name or non-person name (SPRIDEN_SEARCH_LAST_NAME) be included in all your rules.
- For data elements where you can change the number of characters to use in matching from what exists in the data dictionary, you cannot change the length to zero.
- If you enter a negative length for the number of characters to compare for the field, the algorithm will start checking the specified number of characters with the last character in the string.

Note

SunGard Higher Education recommends that you not use a negative length for SSN/SIN/TIN. It could create performance issues. ■

Different pieces of data are required depending on whether you are adding a person or non-person record:

- For a *person* record, data in **Last Name** must exist. No other data is required to initiate the Common Matching process. If data is entered in the **Last Name** field, the GOTCMME_ENTITY_IND must equal *P* for matching a person.
- For a *non-person* record, data in the Non-Person Name only must exist. No other data is required to initiate the Common Matching process. If data is entered in the **Non-person Name** field, the GOTCMME_ENTITY_IND must equal *C* for matching a non-person.

Two-Step Matching Process

The algorithm performs a two-step matching process referred to as Primary Match and Secondary Match on each of the rules associated with the Matching Source to evaluate new identification records that are being added to the Banner database to see if they already exist. For more information on how the Primary and Secondary Match work, refer to [“Primary Match Processing” on page 3-3](#).

Note

When the Algorithm is matching on non-code fields such as Name, all spaces and special characters are removed, and the text is forced to be uppercase. ■

The algorithm will return:

- *New* if no matching records or potential matches are found, based on the rules defined for the Matching Source.
- *Match* if only one record is found that matches all the criteria of the rules. In this case, the algorithm will return the ID as a Match.
- *Potential Match (the incoming new record gets a status of Suspense)* if multiple records are found that match the criteria, or if records are found that match only some of the criteria. In this case, the algorithm returns the IDs of all possible matches. The data is stored in a temporary table until the user finishes investigating them.

For a match or a potential match, the conditions that were met and unmet will also be returned, for example:

Name Match, Date of Birth Year Match, Address No Match

Primary Match Processing

The primary match uses last name/non-person name (SPRIDEN_SEARCH_LAST_NAME), a required data element for Common Matching. If the following items are specified in your rules, then they will be included in the primary match:

- If you specify first name (SPRIDEN_SEARCH_FIRST_NAME) or middle name (SPRIDEN_SEARCH_MI) in a rule, they will be used as part of the primary match for name.
- If you specify SSN/SIN/TIN (SPBPERS_SSN) in a rule, it will be used in the primary match.
- If you specify ID (SPRIDEN_ID) (used only for batch processes) in a rule, it will be used in the primary match.

The algorithm uses the **Entity** indicator defined for the Matching Source on the Common Matching Source Code Rules Form (GORCMSC) to determine the records to select in Banner:

- *P* (person records): GORCMSC_ENTITY_CDE = P will select records from SPRIDEN where SPRIDEN_ENTITY_IND = P.
- *C* (non-person records): GORCMSC_ENTITY_CDE = C will select records from SPRIDEN where SPRIDEN_ENTITY_IND = C.
- *B* (both person and non-person records): GORCMSC_ENTITY_CDE = B will select record from SPRIDEN where SPRIDEN_ENTITY_IND = P or C.

For Online Processing, Either Step 1 or Step 2 below must be true for a record to pass the primary match. If the External record fails the primary match, then the match status will be marked new.

For Batch processing, If the ID is included in the Rule, either Step 1, Step 2 or Step 3 below must be true for a record to pass the primary match.

1. If SSN/SIN/TIN is defined for the Matching Source and rule set number, then all records from Banner with a matching SSN/SIN/TIN will be retrieved.

SSN/SIN/TIN is part of the rule and SPBPERS_SSN = External source SSN/SIN/TIN.

2. If the first name and/or middle names are defined for the Matching Source and rule set number, combine with the last name criteria and retrieve all records from Banner with a matching name.

 **Note**

When matching non-person records, the first and middle names should not be included as part of the rule. ■

- 2.1. The following must be true:

SPRIDEN_SEARCH_LAST_NAME must equal the last name on the external source for the specified length.

 **Note**

If the Matching Source has been defined to look for non-person records and SPRIDEN_SEARCH_LAST_NAME is not like the non-person name from the external source, the matching algorithm will check to see if a matching record exists on the GORNPNM alias table. ■

- 2.2. One of the following must be true:

- First name data element is not defined.

OR

- First name data element is defined for the rule and SPRIDEN_SEARCH_FIRST_NAME is equal to the external source first name for the specified length.

 **Note**

If the SPRIDEN_SEARCH_FIRST_NAME is not like the first name from the external Source, the matching algorithm will check to see if a matching record exists on the GORNAME alias table if the Matching Source is defined to match person records. ■

- 2.3. One of the following must be true:

- Middle name data element is not defined.
- Middle name data element is defined for the rule and SPRIDEN_SEARCH_MI is equal to the external source middle name for the specified length.

 **Note**

If the SPRIDEN_SEARCH_MI is not like the middle name from the external source, the matching algorithm will check to see if a matching record

exists on the GORNAME alias table if the Matching Source is defined to match person records. ■

3. For batch processing where the ID may be provided in a record from an external source, if the ID is defined for the source and rule set number, then retrieve all records from Banner with a matching ID.

ID data element is defined as part of the rule and `SPRIDEN_ID = External source ID`.

Secondary Match Processing

The secondary match will compare the data elements defined for the Matching Source and rule set number for all records returned by the primary match process for the current Rule Set. The goal of this match is to find an 'exact' match between the incoming record and a Banner record.

When comparing a data field with a *Yes* value in the **Match on Null Data Yes or No?** field on the GORCMRL form, a null value may exist either in Banner or the external source. If a null value exists either in Banner or the external source for the data element, the data element is considered as matched.

When the **Match on Null Data Yes or No?** field is *No*, if the field is null in either Banner or the incoming record for the data element, it is not considered during the matching process and a message will be returned that data is missing in the Match Status.

This step is repeated for each of the data elements for the current Rule Set number and one condition must be true for the data element to be considered a match:

- The **Match on Null Data Yes or No?** field is *No*, but the data element is not present on the incoming record or the Banner record.
- The **Match on Null Data Yes or No?** field is either *Yes* or *No* for the data element, and the Banner value is equal to the incoming record value for the specified length.
OR
- The **Match on Null Data Yes or No?** field is *Yes* for the data element and the Banner value is null.
OR
- The **Match on Null Data Yes or No?** field is *Yes* for the data element and the incoming record value is null.

When the data being matched is part of a logical unit (such as when you are matching *City* as part of an address), the logical unit is matched separately and completely. For example,

when matching on city and ZIP code, the city and ZIP code must be associated with one address.

 **Note**

For an incoming record to be considered new when the record has already passed the primary match, all non-name data elements must be determined to be not matched, and none of the non-name elements may be null. ■

Matching Using Multiple Rule Priorities

The Common Matching Algorithm can process multiple rules. Each rule must have a priority number to specify the order in which it is processed. The strictest rule should be the last Rule Set.

The Common Matching procedure will process each rule in order, separately and completely:

1. The algorithm will perform the Primary Matching for the rule using name and/or SSN as defined in the rule to establish the population on which the rest of the processing (secondary match) will be performed.

If it does not find a match, the incoming record is considered *New*.

2. The algorithm will perform the secondary matching processing against the results of the primary match.
 - If the programming logic finds an exact match with only one record, the incoming record is considered a Match.
 - If more than one record is matched to the criteria, the incoming record is considered in *Suspense*. The incoming record is also considered in suspense if the data matches some of the criteria but not all of it.

If the results are New or Match, the results are returned to the user or calling process. No other rules are processed.

3. The algorithm continues evaluating all the rules for the Matching Source, one at a time, separately and completely. When all the rules have been processed (or the incoming record is determined to be new or a Match), the algorithm will return the results to the user or calling process.

The Match Status (*New*, *Matched*, or *Potential Match*) will be returned with a status message specifying which elements were matched, which ones were not matched, and which ones were missing, based on the rules.

 **Note**

The Common Matching algorithm strips special characters out when doing a comparison. If you try to enter a new identification record with the address *123 Maple \$ Street* and there is a record on the database that

has (for the same name) *123 Maple Street*, the algorithm will consider it a match if you set up your rules to check addresses. ■

For each Rule Set, the algorithm performs the two step matching process - the Primary Match and the Secondary Match. The algorithm performs the Primary Match using the name data and/or the SSN/SIN/TIN data as defined in your current rule set. If you include first name and middle name in your rule, then they will be included in the primary match. If you specify SSN/SIN/TIN in your rule, it will also be used in the primary match. The name and SSN/SIN/TIN data are evaluated independently. Therefore the results of the primary match may include records that match only on name and also include records that only match on SSN/SIN/TIN.

Note

For batch processing ID can also be included as an element in your matching rules. If ID is included in the rules it will also be used to find matches during the primary match process of the algorithm. ■

If the algorithm finds potential matches using the Priority 1 Rule, it stores them in the temporary table, and checks for any additional rule priorities associated with the chosen Matching Source.

If the algorithm finds an additional rule (i.e., Rule Set 2), it starts searching the entire database again for a match, using the next rule set's criteria. The algorithm will perform the Primary Match again, this time using the Name and/or SSN/SIN/TIN criteria as defined in Rule Set 2. If it finds any records that pass the primary match on Rule Set 2 it will then perform the Secondary Match on those records to narrow down those records based on the rest of the criteria in Rule Set 2. Any potential matches will be stored in the temporary table, and the algorithm will look for the next rule set (i.e. Rule Set 3).

This process continues until all rules are processed or one of the rules identifies the incoming data to be either New or Matched. Once a rule determines that the record is New or Matched the algorithm returns the results and no further rules are processed.

All the matches and potential matches are then displayed on the appropriate tab on GOAMTCH. It is possible for a record to appear more than once on the Potential Matches tab. This occurs when the record was found as a match by more than one Rule. If you highlight the record the Match Status will indicate which rule the record matched on and what elements matched, did not match or are required and missing.

Multiple Rule Priorities in Online Processing

It is possible for a record to have different statuses due to different rules. A record could be considered a potential match under one rule and a match using one of the subsequent rules.

In those cases, the record's status is Match, but the user can see the potential matches that were discovered using the prior rules on the Common Matching Entry Form (GOAMTCH).

For example, if Rule Set 1 finds potential matches, the new record will get a status of suspense and potential matches will be stored to display on the Potential Matches tab. Then, Rule Set 2 is executed and finds an exact match. The overall status of the new record will be *Match* and the matched record will be displayed on the Match tab on GOAMTCH. The potential matches returned by Rule Set 1 will be listed on the Potential Matches tab, along with any potential matches returned by Rule Set 2.

How Common Matching Creates and Updates Records

When the algorithm determines that the incoming identification record does not already exist, the user can select the **Create New** button on GOAMTCH to create the record in the Banner database.

The form will attempt to insert a record into the following tables using the associated APIs:

API Name	Type of Data Inserted	Updated Table
IDENTIFICATION	Name	SPRIDEN
ADDRESS	Address	SPRADDR
TELEPHONE	Telephone	SPRTELE
EMAIL	E-mail	GOREMAL
BIO	Biographical/Demographic	SPBPERS

Note

If any of the above APIs fail during the creation of new record a failure message will be displayed indicating what API failed. ■

If the algorithm determines that the record already exists on the database, the user can update the record. The Common Matching process updates the tables (except for SPRIDEN) via the API.

The records are created or updated in the following order:

1. Identification (only if a new record is being created)
2. Address
3. Telephone

4. E-mail
5. Biographical

If any of these are missing, the algorithm moves to the next.

Address Record

If the user has entered an address type and address, the programming logic checks to see if one already exists in the database:

- If the address type exists but the actual address is different, the existing address is made inactive and the new address is inserted.
- If the address type doesn't exist, the new address is created.
- If the only address on the database for the address type is inactive, the new address record is created.

Note

The programming logic that checks to see if the addresses are the same is looking for an exact match.

If, for example, you have the address *23 MAPLE STREET* on the database and the user enters *23 MAPLE ST.*, the programming logic will consider them to be different addresses. It will make the existing address inactive, and will insert the new one into the SPRADDR table. ■

If the user hasn't entered some of the required values to create or update an address (e.g., City), the address record will not be created and the programming logic will begin processing the telephone information. The status message will indicate that the address record create or update failed (depending on whether you selected the **Create New** or **Update ID** buttons, respectively).

Telephone Record

If the user has entered a telephone type and number, the programming logic checks to see if one already exists in the database:

- If there is, the programming logic will begin processing e-mail address information.
- If there isn't, the new record will be inserted into SPRTELE.

If the address record is being created at the same time, SPRTELE_ATYP_CODE and SPRTELE_ADDR_SEQNO will be populated with the same address type and sequence number that are being populated in SPRADDR.

If the address record is not being created at the same time, `SPRTELE_ATYP_CODE` and `SPRTELE_ADDR_SEQNO` will be null on the newly-created record.

E-mail Record

If the user enters an e-mail address type and e-mail address, the programming logic will check to see if there is an exact match for that type. It does not consider case. (If an e-mail address exists in the database in lowercase and you try to add it in uppercase, the change will not be made.)

- If there is a match, the programming logic displays a status message.
- If there isn't, the new record will be inserted into GOREMAL.

When you create a new e-mail address for a type, the **Preferred** indicator is set to *N*.

If you try to add an e-mail address and it already exists but has been marked inactive, the new address will not be created. Adding the address again will not make it active.

Note

A primary key in the GOREMAL table enables the APIs to process e-mail information properly. ■

Biographical Record

To match the person in the top half of the form to the person found by the match, choose either:

- **Select ID** to select the record and carry it back to the key block of the %IDEN form.
- **Update ID** to update the record with data from top block.

Clicking **Update ID** will update the following data, if it has been entered in the Data Entry block:

- SSN/SIN/TIN if it is null in Banner
- Date of Birth if it is null in Banner
- Gender if it is Unknown in Banner
- Address, telephone, and e-mail if the type does not exist for the record.
- A new sequence for the address will be created if the same type exists but address information is different.

The Status Message

After you create a new identification record or update an existing one, the programming logic displays a status message in a pop-up window.

It lists types of record (identification, biographical information, address, telephone, and e-mail) and indicates whether or not any of the records were created or changed.

When you are creating a new record, the valid statuses are:

- *Created*
- *Not Created*

When you are updating a record, the valid statuses are:

- *Updated*
- *Not Updated*
- *Created*
- *Not Created*

For example, if you changed an existing address and added gender information in the Data Entry block of GOAMTCH, the message would say:

*Biographical record created; Address record created;
Telephone not updated. E-mail not updated.*

If any of the APIs fail the status message will indicate which API failed. An example of an API failure would be that an address type was entered but no address information was entered. The system cannot create an Address record with missing information.



4 Implementation Examples

This chapter describes various Common Matching scenarios and the results they produce.

Sample Common Matching Scenarios

The following examples demonstrate some different scenarios that may occur when using Common Matching. These examples are for informational purposes only.

 **Note**

If you cannot see information that you think you should see, check your Fine-Grained Access (FGAC) rules for Value-Based Security and PII. FGAC rules may prohibit you from seeing certain data. ■

Person Search That Results in Many Potential Matches

This situation occurs when the rule is not strict enough to rule out many potential matches. The following is an example of one condition that would cause this result:

Rule, Priority 1

- Last Name, 5 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- First Name, 1 character, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- E-mail, 90 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*

Data Input on GOAMTCH

Last Name: *Smith*
First Name: *S*

Records Found and Why

Smith, Sally
Smith, Sandy
Smith, Sara
Smithson, Suzanne
Smithingdale, Scott

The above (as well as many other records) are displayed as potential matches on Rule 1 with Match Results of 'Name Match, E-mail Match'.

The *Primary Match* identifies all records with a name match. In this all case, all records with a last name beginning with *Smith* and a first name beginning with *S*.

The *Secondary Match* process then attempts to narrow down the pool of identified potential matches by looking for matching e-mail records.

Since the E-mail data element is marked as *Yes* in the Rule null values on GOAMTCH, null values in Banner or null values in both count as a potential match.

Since an e-mail address was not entered on GOAMTCH, any records in Banner which have a blank e-mail address and any records in Banner with an e-mail address present will count as a potential match.

In this case, all the records identified by the primary match will be returned as potential matches.

Person Search That Results in One Match and Potential Matches

This situation occurs when one record matches the rule exactly but other are considered potential matches. The following is an example of one condition that would cause this result:

Rule, Priority 1

- Last Name,
5 characters,
Match on Null Data Yes or No? field on GORCMRL set to *No*
- First Name,
1 character,
Match on Null Data Yes or No? field on GORCMRL set to *No*
- City,
5 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- E-mail,
90 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*

Data Input on GOAMTCH

Last Name: *Smith*
First Name: *S*
City: *Berwyn*

Records Found and Why

Record Displayed on Match Tab:

Smith, Sandy with an address that contains the City *Berwyn*

Match Results: Name Match, Address Match, E-Mail Match

This record displays on the Match Tab because there are no other records in the database that have a City beginning with *Berwy* and a Last Name beginning with *Smith* and a First Name beginning with *S*.

Records Displayed on the Potential Matches Tab:

Smith, Sally

Smith, Sandy

Smith, Sara

Smithson, Suzanne

Smithingdale, Scott

The above (as well as many other records) are displayed as potential matches on Rule Set 1 with Match Results of 'Name Match, Address Missing, E-mail Match'.

The *Primary Match* identifies all records with a name match. In this all case, all records with a last name beginning with *Smith* and a first name beginning with *S*.

The *Secondary Match* process then attempts to narrow down the pool of identified potential matches by looking for matching address records and/or matching E-mail Records.

All the records displayed on the potential match tab are records that:

- Do not have any address record in the Banner but are considered a match on e-mail (status message will read 'Address Missing, E-mail Match'). Since the city element is marked as *Required* in the Rule, if it is null in either GOAMTCH or Banner the element will not be considered for matching and the Match Result message will indicate that required data is missing.
- Have an address record whose City does not begin with *Berwy* but are considered a match on e-mail (status message will read 'Address No Match, E-mail Match'). Since the E-mail data element is marked as *Exists* in the Rule, null values on GOAMTCH, null values in Banner or null values in both count as a potential match. (Since an e-mail address was not entered on GOAMTCH any records in Banner which have a blank e-mail address and any records in Banner with an e-mail address present will count as a potential match.)

Person Search That Results in an Exact Match

In this example, multiple records pass the Primary Match and are then narrowed down to one match.

Rule, Priority 1

- Last Name, 8 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- First Name, 4 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- E-mail, 10 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*

Data Input on GOAMTCH

Last Name: *Johnston*

First Name: *Edward*

E-mail: *ejohnston@aol.com*

Records Found and Why

The *Primary match* identifies the following as potential matches:

- *Edward Johnston*
- *Edwardo Johnston*
- *Edward Johnstonling*

The *secondary match* narrows the previous potential matches to one match, based on the e-mail record:

- *Edward Johnston*, who has an e-mail address of *ejohnston@aol.com*.

The following records are presented as *potential matches*:

- *Edwardo Johnston*
- *Edward Johnstonling*

Two potential matches are returned because the Priority 1 rule is met, but the e-mail addresses, although present, do not match.

Status is New but Potential Matches Exist from a Prior Rule

In this example, the Match Status is New, but Potential Matches exist from Prior rule. This situation occurs when the record is considered a potential match by Rule Set 1, and then determined to be New by Rule Set 2.

Below is an example of one condition that would cause this result:

Rule, Priority 1

- Last Name, 5 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- First Name, 1 character, Required
- E-mail, 90 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*

Rule, Priority 2

- Last Name, 5 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- First Name, 3 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*

Data Input on GOAMTCH

Last Name: *Smith*
First Name: *Sue*
E-mail: *ssmith@aol.com*

Records Found and Why

Smith, Sally
Smith, Sandy
Smith, Sara

The above all are displayed as potential matches on Rule 1 with Match Results of 'Name Match, E-mail Match'. None of the above records have an e-mail address on their Banner record, however, since the E-mail element is marked *Yes* and it is being provided on GOAMTCH, it is considered a match.

Rule Set 2 considers the input data to be New because both the First Name and Last Name are not an exact match.

Match Results That Show a Value Missing

The following status is returned when elements in the rule have the **Match on Null Data Yes or No?** field on GORCMRL set to *No*, but have not been supplied on GOAMTCH:

Rule, Priority 1

- Last Name, 5 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- First Name, 3 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- E-mail, 90 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*

Rule, Priority 2

- Last Name, 5 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*
- E-mail, 3 characters, **Match on Null Data Yes or No?** field on GORCMRL set to *No*

Data Input on GOAMTCH

Last Name: *Smith*

First Name: *Bil*

Records Found and Why

Potential Matches:

- *Billy Smithy*
Match Priority 1
Name Match, E-mail Match
(This record is considered a potential match on Rule Set 1 because the e-mail element is not required, and has not been input on GOAMTCH. Elements that are marked as *Yes* in a Rule and are Null in Banner or GOAMTCH or both are considered a match.)
- *Abigail Smith*
Match Priority 2
Name Match, E-mail Missing
- *Alice Smith*
Match Priority 2
Name Match, E-mail Missing
- *Billy Smithy*
Match Priority 2
Name Match, E-mail Missing

(These 3 records match on Rule Set 2 because the e-mail element is required in Rule Set 2 and the e-mail address has not been provided on GOAMTCH. The records pass the primary match on Rule Set 2 and are marked as potential matches because of the missing required element. The Match Results message will indicate that data is missing.)

Data Elements Required but Not Provided

In this example, the incoming record is suspended because all non-name Required Data Elements are missing on GOAMTCH.

Rule

- Last Name:Required (**Match on Null Data Yes or No?** field on GORCMRL set to *No*)
- First Name = Required (**Match on Null Data Yes or No?** field on GORCMRL set to *No*)
- DOB Day = Required (**Match on Null Data Yes or No?** field on GORCMRL set to *No*)
- DOB Month = Required (**Match on Null Data Yes or No?** field on GORCMRL set to *No*)
- DOB Year = Required (**Match on Null Data Yes or No?** field on GORCMRL set to *No*)
- City = Required (**Match on Null Data Yes or No?** field on GORCMRL set to *No*)

Data Input on GOAMTCH

Name: *Mildred Jones*

Records Found and Why

Mildred Jones, DOB = 08/17/1957, City = Topeka

The incoming record will pass the Primary Match because the first and last names match. However, since all other data elements are missing (i.e. they have not been provided on GOAMTCH or are not provided on the incoming tape or dataload) they are considered not matched but null and the system put the incoming record in suspense and will display the Potential Match found.

Data Elements Set as Exists Are Provided and Do Not Match

In this example, the incoming record passes the primary match as the first and last name matches against 2 Banner records. These 2 records are then used in the secondary match. One record is considered a potential but the other is ruled out by the secondary match.

Rule Set 1:

- Last Name,
- **Match on Null Data Yes or No?** field on GORCMRL set to *No*;
- First Name, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- SSN, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- DOB, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- City, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- Zip Code, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;

Rule Set 2:

- Last Name, **Match on Null Data Yes or No?** field on GORCMRL set to *No*;
- First Name, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- DOB, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- City , **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- Zip Code, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;

Data Input on GOAMTCH

Alberta Rockville,

330229101,

Largesse,

06259,

no DOB

Records Found and Why

Alberta Rockville, 330229101, Largesse, 06259, 05/01/1985

The incoming record will pass the Primary Match because the first and last name matches at least one Banner record with the same first and last name. Using Rule 1

and the Matching Algorithm, the external record will match against the above Banner record.

However, even though the record below passed the Primary Match on Rule Set 1 and Rule Set 2, it was not returned as a potential match because the City and Zip Code were provided on GOAMTCH and compared to the values existing in Banner and did not match.

Alberta Rockville, no SSN, Pomfret, 19355, no DOB

Only Exception to the Basic Matching Algorithm

In this example, the external record passes the primary match, which usually means that the match status will be *Suspense* at a minimum. However, in this case, because none of the non-name/SSN fields match, the external record is set as *New*.

Rule Set 1

- Last Name, **Match on Null Data Yes or No?** field on GORCMRL set to *No*;
- First Name, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- SSN, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- DOB, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- City, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- Zip, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;

Rule Set 2

- Last Name, **Match on Null Data Yes or No?** field on GORCMRL set to *No*;
- First Name, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- City, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;
- Zip, **Match on Null Data Yes or No?** field on GORCMRL set to *Yes*;

Data Input on GOAMTCH

Tomasso Dalimonte

SSN = null,

DOB = 09/07/59,

City = Woodstock,

ZIP = 06281

Records Found and Why

Tomasso Dalimonte,
SSN = null,
DOB = 06/02/78,
City = Marikesh,
ZIP = 11233

The external record will pass the primary match because the first and last names match. Normally, this would mean that the record would be suspended at a minimum. However, because the DOB, City and ZIP code fields specifically do not match (i.e., none of them are null), then the record's match status is set to New. This is the only exception to the basic matching algorithm.

Sample Rules

Banner Finance Rule for Vendors (Persons)



Tip

To use this rule for Non-Persons, you should not include the First Name element.

Priority 1:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	8	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	6	Yes
SPBPERS_SSN	SSN/SIN/TIN	9	Yes

Priority 2:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	8	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	6	Yes

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRADDR_CITY	CITY	6	Yes
SPRADDR_STAT_CODE	STATE	2	Yes
SPRADDR_ZIP	ZIP/POSTAL CODE	3	Yes
SPRTELE_PHONE_AREA	AREA CODE	3	Yes
SPRTELE_PHONE_NUMBER	TELEPHONE NUMBER	4	Yes

Banner Advancement Rule for Persons



Tip

To use this rule for Non-Persons, you should not include the First Name Element.

Priority 1:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	60	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	5	No
SPRIDEN_ID	ID	9	No

Priority 2:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	20	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	1	No

Priority 3:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	5	No
SPRADDR_STREET_LINE1	STREET LINE 1	15	Yes
SPRADDR_CITY	CITY	10	Yes

Priority 4:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	5	No
SPRADDR_CITY	CITY	10	Yes
SPRTELE_PHONE_NUMBER	TELEPHONE NUMBER	7	Yes

Priority 5:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	5	No
SPRADDR_CITY	CITY	3	Yes
SPRADDR_ZIP	ZIP/POSTAL CODE	5	Yes

Banner Student Rule

Priority 1:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	10	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	10	No
SPBPERS_SSN	SSN/SIN/TIN	9	No

Priority 2:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	10	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	10	No
SPBPERS_SSN	SSN/SIN/TIN	9	Yes
SPBPERS_BIRTH_DAY	DATE OF BIRTH DAY	2	Yes
SPBPERS_BIRTH_MONTH	DATE OF BIRTH MONTH	2	Yes
SPBPERS_BIRTH_YEAR	DATE OF BIRTH YEAR	2	Yes
SPRADDR_CITY	CITY	10	Yes
SPRADDR_ZIP	ZIP/POSTAL CODE	5	Yes

Priority 3:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	10	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	10	No
SPBPERS_BIRTH_DAY	DATE OF BIRTH DAY	2	Yes
SPBPERS_BIRTH_MONTH	DATE OF BIRTH MONTH	2	Yes
SPBPERS_BIRTH_YEAR	DATE OF BIRTH YEAR	2	Yes
SPRADDR_CITY	CITY	10	Yes
SPRADDR_ZIP	ZIP/POSTAL CODE	5	Yes

Banner Financial Aid Rule for Persons

Priority 1:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	10	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	10	No
SPBPERS_SSN	SSN/SIN/TIN	9	No
SPBPERS_BIRTH_DAY	DATE OF BIRTH DAY	2	Yes
SPBPERS_BIRTH_MONTH	DATE OF BIRTH MONTH	2	Yes

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPBPERS_BIRTH_YEAR	DATE OF BIRTH YEAR	2	Yes
SPRADDR_CITY	CITY	10	No

Priority 2:

Column Name	Data Element	Length	Match on Null Data Yes or No? field on GORCMRL set to Yes or No
SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	10	No
SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	10	No
SPBPERS_BIRTH_DAY	DATE OF BIRTH DAY	2	Yes
SPBPERS_BIRTH_MONTH	DATE OF BIRTH MONTH	2	Yes
SPBPERS_BIRTH_YEAR	DATE OF BIRTH YEAR	2	Yes
SPRADDR_CITY	CITY	10	No

Sample Banner Advancement Scenarios

Advancement Rule For Person Matching

Rule Priority	Column	Element	Length	Match on Null Data
Priority 1	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	60	Yes
	SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	5	Yes
Priority 2	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON- PERSON NAME	20	Yes
	SPRIDEN_SEARCH_FIRST_NAME	FIRST NAME	5	Yes

Rule Priority	Column	Element	Length	Match on Null Data
Priority 3	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	10	Yes
	SPRADDR_STREET_LINE1	STREET LINE 1	15	Yes
	SPRADDR_CITY	CITY	10	Yes
Priority 4	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	10	Yes
	SPRADDR_CITY	CITY	10	Yes
	SPRTELE_PHONE_NUMBER	TELEPHONE NUMBER	7	Yes
Priority 5	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	10	Yes
	SPRADDR_CITY	CITY	10	Yes
	SPRTELE_PHONE_AREA	TELEPHONE AREA CODE	3	Yes

Additional Rule Settings

You may use the following additional rules, if desired. Note that they may affect performance.

- Add comments for your rule to better assist users in selection for match process (on GORCMRL).
- Check the **Transpose First/Last Name** check box in the Options block on GORCMSC to allow the search to look for name-entered matches on either first or last name.

Note

If entering only one part of the name, enter the details in the last name field and the search process will look for a match in both the first and last name fields. ■

- Check the **Allow Alias Wildcard Use** check box in the Options block on GORCMSC to allow the search to use user-defined alias values (as specified in GORNAME) in search match selection.
- Check the **Allow Length Override** check box in the Options block on GORCMSC to allow the search process to use the number of characters entered for selection, even if less are entered than defined on the rule (on GORCMRL).
- Check the **Prevent ID Creation on API Failure** check box in the Options block on GORCMSC to avoid creating a record in error when missing required data fields are not populated on GOAMTCH (requires complete Address, Telephone and/or E-mail details in order to create a new ID).
- Consider defining a hierarchy for address display (under *Hierarchy of Display* on GORCMSC) which results in the display of existing addresses based on the hierarchy defined even if the existing address does not match the search criteria entered. You can use this feature for telephone and e-mail as well.

Advancement Person Examples

The match process follows the rule set order and searching stops when a potential match is found. Define the priority details to begin with the most restrictive details and then broaden the search details in higher numbered priorities.

Existing Data in Banner

First Name	Last Name	Street 1	City	Telephone
Margaret	Chillicothoniannism	222 Spruce Street	Malvern	610-777-9999
Mary	Jones	222 Spruceset Court	Galveston	711-888-4444
John	Smith	1 Fort Way	Atlanta	451-844-5555

Case#1

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Chillicothoniannism	LAST NAME/NON-PERSON NAME	60	Yes
	Margaret	FIRST NAME	5	Yes
Results: Match on priority 1 rule - exact match on first and last name. Stopped after first priority rule based on match status.				

Case #2

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Chillicothoniannisms	LAST NAME/NON-PERSON NAME	60	Yes
	Margaret	FIRST NAME	5	Yes
Results: No match on priority 1 rule - different name				
Priority 2	Chillicothoniannisms	LAST NAME/NON-PERSON NAME	20	Yes
	Margaret	FIRST NAME	5	Yes
Results: Match on priority 2 rule - match on first and last name (based on length). Stopped after second priority rule based on match status.				

Case #3

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Chillicothonianisms	LAST NAME/NON-PERSON NAME	60	Yes
	Margaret	FIRST NAME	5	Yes
Results: No match on priority 1 rule - different name.				

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 2	Chillicothonianisms	LAST NAME/NON-PERSON NAME	20	Yes
	Margaret	FIRST NAME	5	Yes
Results: No match on priority 2 rule - different name.				
Priority 3	Chillicothonianisms	LAST NAME/NON-PERSON NAME	10	Yes
	222 Spruce Street	STREET LINE 1	15	Yes
	Malvern	CITY	10	Yes
Results: Match on priority 3 rule - match on last name (based on length), street, and city Stopped after third priority rule based on match status.				

Case#4

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Chillicothonianisms	LAST NAME/NON-PERSON NAME	60	Yes
	Margaret	FIRST NAME	5	Yes
Results: No match on priority 1 rule - different name.				
Priority 2	Chillicothonianisms	LAST NAME/NON-PERSON NAME	20	Yes
	Margaret	FIRST NAME	5	Yes
Results: No match on priority 2 rule - different name.				
Priority 3	Chillicothonianisms	LAST NAME/NON-PERSON NAME	10	Yes
	22 Spruce Street	STREET LINE 1	15	Yes
	Malvern	CITY	10	Yes
Results: No match on priority 3 rule - street is not a match.				
Priority 4	Chillicothonianisms	LAST NAME/NON-PERSON NAME	10	Yes
	Malvern	CITY	10	Yes
	610-777-9999	TELEPHONE NUMBER	7	Yes
Results: Match on priority 4 rule - match on last name (based on length), city, and phone. Stopped after fourth priority rule based on match status.				

Case#5

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Chillicothonianisms	LAST NAME/NON-PERSON NAME	60	Yes
	Margaret	FIRST NAME	5	Yes
Results: No match on priority 1 rule - different name.				

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 2	Chillicothonianisms	LAST NAME/NON-PERSON NAME	20	Yes
	Margaret	FIRST NAME	5	Yes
Results: No match on priority 2 rule - different name.				
Priority 3	Chillicothonianisms	LAST NAME/NON-PERSON NAME	10	Yes
	22 Spruce Street	STREET LINE 1	15	Yes
	Malvern	CITY	10	Yes
Results: No match on priority 3 rule - street is not a match.				
Priority 4	Margaret	LAST NAME/NON-PERSON NAME	10	Yes
	Malvern	CITY	10	Yes
	610-777-9998	TELEPHONE NUMBER	7	Yes
Results: No match on priority 4 rule - telephone is not a match.				
Priority 5	Chillicothonianisms	LAST NAME/NON-PERSON NAME	10	Yes
	Malvern	CITY	10	Yes
	610	TELEPHONE AREA CODE	3	Yes
Results: Match on priority 5 rule - match on last name (based on length), city, and area code. Stopped after fifth priority rule based on match status.				

Advancement Rule For Non-Person Matching

Rule Priority	Column	Element	Length	Match on Null Data
Priority 1	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	60	Yes
Priority 2	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	20	Yes
	SPRADDR_STREET_LINE1	STREET LINE 1	15	Yes
	SPRADDR_CITY	CITY	10	Yes
Priority 3	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	10	Yes
	SPRADDR_CITY	CITY	10	Yes
	SPRTELE_PHONE_NUMBER	TELEPHONE NUMBER	7	Yes
Priority 4	SPRIDEN_SEARCH_LAST_NAME	LAST NAME/NON-PERSON NAME	10	Yes
	SPRADDR_CITY	CITY	10	Yes
	SPRTELE_PHONE_AREA	TELEPHONE AREA CODE	3	Yes

Additional Rule Settings

You may use the following additional rules, if desired. Note that they may affect performance.

- Add comments for your rule to better assist users in selection for match process (on GORCMRL).
- Check the **Allow Alias Wildcard Use** check box in the Options block on GORCMSC to allow search to use user defined alias values (on GORNPMM) in search match selection.
- Check the **Allow Length Override** check box in the Options block on GORCMSC to allow the search process to use the number of characters entered for selection, even if less are entered than defined on the rule (as specified in GORCMRL).
- Check the **Prevent ID Creation on API Failure** check box in the Options block on GORCMSC to avoid creating a record in error when missing required data fields are not populated on GOAMTCH (requires complete Address, Telephone and/or E-mail details in order to create a new ID).
- Consider defining a hierarchy for address display (under *Hierarchy of Display* on GORCMSC) which results in the display of existing addresses based on the hierarchy defined even if the existing address does not match the search criteria entered. You can use this feature for telephone and e-mail as well.

Advancement Non-Person Examples

The match process follows the rule set order and searching stops when a potential match is found. Define the priority details to begin with the most restrictive details and then broaden the search details in higher numbered priorities.

Existing Data in Banner

Name	Street 1	City	Telephone
Michael Tire	444 Tires Street	Malvern	610-444-9999
Scott Oil	14 Summer Court	Galveston	711-777-4444
Peterson Book Binders	17 Reading Road	Atlanta	451-944-5555

Case #1

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Michael Tire	LAST NAME/NON-PERSON NAME	60	Yes

Results: Match on priority 1 rule - exact match on name. Stopped after first priority rule based on match status.

Case#2

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Peterson Book Binders Inc	LAST NAME/NON-PERSON NAME	60	Yes
Results: No match on priority 1 rule - different name.				
Priority 2	Peterson Book Binders Inc	LAST NAME/NON-PERSON NAME	20	Yes
	17 Reading Road	STREET_LINE1	15	Yes
	Atlanta	CITY	10	Yes
Results: Match on priority 2 rule - match on name (based on length), street, and city. Stopped after second priority rule based on match status.				

Case#3

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Peterson Books Binders Inc	LAST NAME/NON-PERSON NAME	60	Yes
Results: No match on priority 1 rule - different name.				
Priority 2	Peterson Books Binders Inc	LAST NAME/NON-PERSON NAME	20	Yes
	17 Reading Road	STREET_LINE1	15	Yes
	Atlanta	CITY	10	Yes
Results: No match on priority 2 rule - different name.				
Priority 3	Peterson Books Binders Inc	LAST NAME/NON-PERSON NAME	10	Yes
	Atlanta	CITY	10	Yes
	944-5555	TELEPHONE	7	Yes
Results: Match on priority 3 rule - match on name (based on length), city, and telephone. Stopped after third priority rule based on match status.				

Case#4

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 1	Peterson Books Binders Inc	LAST NAME/NON-PERSON NAME	60	Yes
Results: No match on priority 1 rule - name is different.				
Priority 2	Peterson Books Binders Inc	LAST NAME/NON-PERSON NAME	20	Yes
	17 Reading Road	STREET_LINE1	15	Yes
	Atlanta	CITY	10	Yes
Results: No match on priority 2 rule - name is different.				

Rule Priority	Entered Data to Match	Element	Length	Match on Null Data
Priority 3	Peterson Books Binders Inc	LAST NAME/NON-PERSON NAME	10	Yes
	Atlanta	CITY	10	Yes
	945-5555	TELEPHONE	7	Yes
Results: No match on priority 3 rule - telephone is not a match (one wrong number).				
Priority 4	Peterson Books Binders Inc	LAST NAME/NON-PERSON NAME	10	Yes
	Malvern	CITY	10	Yes
	451	TELEPHONE AREA CODE	3	Yes
Results: Match on priority 4 rule - match on last name (based on length), city, and area code. Stopped after fourth priority rule based on match status.				

Converting Legacy Data - FGAC and Common Matching

When you are converting legacy data, it is very important that you turn off Value-Based Security (VBS) and PII security for Common Matching, and then turn it back on when the conversion process is finished. Failure to do this may have a severe impact on performance.

Warning

The user who connects to the Banner database must be exempt from PII before running the Common Matching package. VBS and PII are described in detail in the *Banner Data Security Handbook*.

Integrating Common Matching into Site-Specific Processes

The following steps describe how to integrate Common Matching into batch processes at your institution.

Step 1 Insert Data into GOTCMME

Insert Data into the Common Matching Entry Table (GOTCMME), which is an Oracle global temporary table.

Use the `P_INSERT_GOTCMME` procedure in the Common Matching package (GOKCMPL) to enter all the data into the Common Matching Entry Table (GOTCMME). Do not be concerned about the Common Matching rules.

```

PROCEDURE p_insert_gotcmme
(
  p_last_name      IN  gotcmme.gotcmme_last_name%TYPE,
  p_entity_cde    IN  gotcmme.gotcmme_entity_cde%TYPE,
  p_first_name    IN  gotcmme.gotcmme_first_name%TYPE,
  p_mi            IN  gotcmme.gotcmme_mi%TYPE,
  p_id            IN  gotcmme.gotcmme_id%TYPE,
  p_street_line1  IN  gotcmme.gotcmme_street_line1%TYPE,
  p_street_line2  IN  gotcmme.gotcmme_street_line2%TYPE,
  p_street_line3  IN  gotcmme.gotcmme_street_line3%TYPE,
  p_city          IN  gotcmme.gotcmme_city%TYPE,
  p_stat_code     IN  gotcmme.gotcmme_stat_code%TYPE,
  p_zip          IN  gotcmme.gotcmme_zip%TYPE,
  p_natn_code     IN  gotcmme.gotcmme_natn_code%TYPE,
  p_cnty_code     IN  gotcmme.gotcmme_cnty_code%TYPE,
  p_phone_area   IN  gotcmme.gotcmme_phone_area%TYPE,
  p_phone_number  IN  gotcmme.gotcmme_phone_number%TYPE,
  p_phone_ext     IN  gotcmme.gotcmme_phone_ext%TYPE,
  p_ssn          IN  gotcmme.gotcmme_ssn%TYPE,
  p_birth_day     IN  gotcmme.gotcmme_birth_day%TYPE,
  p_birth_mon     IN  gotcmme.gotcmme_birth_mon%TYPE,
  p_birth_year    IN  gotcmme.gotcmme_birth_year%TYPE,
  p_sex          IN  gotcmme.gotcmme_sex%TYPE,
  p_email_address IN  gotcmme.gotcmme_email_address%TYPE,
  p_atyp_code     IN  gotcmme.gotcmme_atyp_code%TYPE,
  p_tele_code     IN  gotcmme.gotcmme_tele_code%TYPE,
  p_emal_code     IN  gotcmme.gotcmme_emal_code%TYPE
) ;

```

Example

```

gokcmpk.p_insert_gotcmme ( p_last_name      => 'Heston',
                           p_first_name     => 'James',
                           p_mi            => 'Tim',
                           p_entity_cde    => 'P' ) ;

```

The following columns are used by the GOAMTCH form to add or change address, telephone, and e-mail address when the PIDM has already been created for the record. The Common Matching algorithm (the programming logic that compares records looking for a match) ignores these columns:

- GOTCMME_STREET_LINE2
- GOTCMME_STREET_LINE3
- GOTCMME_PHONE_EXT
- GOTCMME_ATYP_CODE
- GOTCMME_TELE_CODE
- GOTCMME_EMAL_CODE

The data in GOTCMME is specific to a session; the rows inserted by a session are deleted when that session is finished. More than one session can use GOTCMME at the same time and each session can only see the rows it created.

Step 2 Execute the Common Matching Procedure

Note

The Common Matching Source Code (p_cmssc_code) must already exist in the Common Matching Rules Table (GORCMSR). ■

```

PROCEDURE p_common_matching
( p_cmesc_code      IN  gorcmsr.gorcmsr_cmesc_code%TYPE,
  p_match_status_out OUT gotcmrt.gotcmrt_result_ind%TYPE,
  p_match_pidm_out  OUT gotcmrt.gotcmrt_pidm%TYPE ) ;

```

There are three possible match statuses:

- The record is *new* (GOTCMRT_RESULT_IND = N). No match has been found on the database based on the rules you set up for that Matching Source code. The procedure will not return a PIDM.
- A *match* (GOTCMRT_RESULT_IND = M) is found for the record. Common Matching has found one, and only one, Banner record that matches the record based on the rules. The procedure returns the PIDM to identify the entry.
- *Suspense* (GOTCMRT_RESULT_IND = S) is the match status if the procedure has found at least one record where some of the fields identified in the rule match the record being entered, but not all. A Banner PIDM is not returned.

The detailed results of the comparison are stored in the Common Matching Results Table (GOTCMRT), an Oracle global temporary table.

As with GOTCMME, this table stores data for a specific session and the data is deleted when the session is finished. Multiple sessions can exist concurrently and each one can only see its own rows.

Step 3 Process the Results of the Common Matching Procedure

Perform the process-specific logic necessary for your site-specific procedure. You can use entries from GOTCMRT as needed (e.g., for reporting purposes), but the rows for the session are deleted from GOTCMRT when the session is finished.

Invoking Common Matching from Additional Forms

As delivered, GOAMTCH cannot be invoked from all forms. A new record was added to the GURUPRF table for the BASELINE user that lists the valid forms:

Column Name	Value
GURUPRF_ID	<i>BASELINE</i>
GURUPRF_GROUP	<i>CM</i>
GURUPRF_KEY	<i>LIST</i>
GURUPRF_STRING	<i>FORMS</i>

If you want to expand its availability, update that record to list the additional forms.

Temporary Access to Restricted Records

Banner's Value-Based Security (VBS) and Personally Identifiable Information (PII) features restrict access to data in the Banner database. These features are described in more detail in the *Banner Data Security Handbook*.

There are four forms on which a user can have temporary access to an ID:

1. Finance Person Search Form (GOIIDEN)
2. Common Matching Entry Form (GOAMTCH)
3. SSN/SIN Alternate Search Form (GUIALTI)
4. Person Search Form (SOAIDEN)

If users call one of these forms from another form, they can see all available records, no matter what their PII restrictions are. They can select a record that they would not normally have had access to and bring it back to the original form.

Note

The Cross Domain Search Indicator for Personal Identification Information checkbox must be selected for this user on the FGAC Person User Defaults Form (GOAFPUD). ■



Index

A

aliases **2-11**
APIs
 APIs in Banner General **1-8**

B

Banner forms **2-16**
batch process review forms **1-7**
batch processes **1-6, 1-7**

C

Common Matching
 batch processing **1-6, 1-7**
 setup steps **2-1**
 when creating an ID record online **1-4**
Common Matching Data Dictionary Form
 (GORCMDD) **2-3**
Common Matching Source Code Rules Form
 (GORCMSC) **2-2**
Common Matching Source Code Validation
 Form (GTVCMSC) **2-2**

D

data dictionary **2-3**
 site-specific elements **2-4**

E

Electronic Application Process Form
 (SAAEAPS) **1-7**
Electronic Application Verify/Load Process
 (SARETMT) **1-7**
Electronic Prospect Inquiry Form (SRIPREL)
 1-7
Electronic Prospect Match Process
 (SRRSRIN) **1-7**

F

fields and forms
 reference information **2-16**
Financial Aid Data Load Part 2 Process
 (RCPMTCH) **1-7**
Financial Aid Suspended Record
 Maintenance Form (RCRSUSP) **1-7**
forms **2-16**
 batch process review **1-7**
 that use Common Matching **1-3**

G

General forms
 reference information **2-16**
GORCMDD **2-3**
GORCMSC **2-2**
GORNAME **2-12**
GORNPNM **2-12**
GTVCMSC **2-2**

M

Matching Source Codes **2-2**

N

Name Translation Rules Form (GORNAME)
 2-12
name translations **2-11**
Non-Person Name Translation Rules Form
 (GORNPNM) **2-12**

O

One-Time Payment Form (PEA1PAY) **1-3**
Online Help **2-16**
Online Transcripts Activity List Form
 (SHAEDIS) **1-7**

P

PEA1PAY 1-3

R

RCPMTCH 1-7

RCRSUSP 1-7

S

SAAEAPS 1-7

SARETMT 1-7

SHAEDIS 1-7

Source Code rules 2-2

SRIPREL 1-7

SRRSRIN 1-7

system functions/administration procedures

 Common Matching Procedures 1-1

T

transposed names 2-2







4 Country View Road
Malvern, Pennsylvania 19355
United States of America
www.sungardhe.com